IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

# AUTOMATION OF THREE-WAY MATCHING FOR BILL PAYMENT USING LARGE LANGUAGE MODELS

**Manasa Varsini S**
B.Sc Data Science, Kumaraguru College of Liberal Arts and Science
**Dr. Emimanimcy**
Assistant Professor, Kumaraguru College of Liberal Arts and Science

**Abstract:** The accounts payable process in most manufacturing and service organizations still relies heavily on manual verification of financial documents, which is both time-consuming and prone to errors. Three-way matching, which involves cross-verifying purchase orders (POs), goods receipt notes (GRNs), and invoices before authorizing payment, is a critical control mechanism in procurement workflows. However, the diversity in document formats and the presence of unstructured data make automation of this process a persistent challenge. This paper proposes a system that leverages Large Language Models (LLMs) to automate three-way matching for bill payment verification. The proposed approach uses LLMs in combination with Optical Character Recognition (OCR) to extract relevant fields from documents in varying formats, including scanned PDFs and images. Once extracted, the data from POs, GRNs, and invoices is systematically compared to identify discrepancies in quantities, pricing, item descriptions, and totals. Bills are then classified as payable or non-payable based on predefined tolerance thresholds. The system also generates real-time dashboards and audit-ready reports for finance teams. By reducing manual intervention, this approach aims to minimize payment errors, speed up invoice processing cycles, and strengthen compliance across the procure-to-pay pipeline. The results demonstrate that LLM-based extraction and matching significantly outperforms traditional rule-based systems in handling diverse and unstructured document formats, making it a scalable solution for enterprise-level accounts payable operations.

**Index Terms:** Three-Way Matching, Large Language Models, Accounts Payable Automation, Optical Character Recognition, Invoice Verification, Procurement, Natural Language Processing.

## I. INTRODUCTION

In the modern business landscape, the accounts payable (AP) function serves as a cornerstone of financial operations for organizations of all sizes. Every time a company purchases raw materials, services, or any form of goods, a chain of financial documents is generated that must be verified before payment is released. This verification process, commonly known as three-way matching, involves comparing three distinct documents: the purchase order (PO) issued by the buyer, the goods receipt note (GRN) confirming delivery, and the invoice submitted by the supplier. The goal is straightforward: to confirm that what was ordered, what was received, and what is being billed all line up correctly [2].

Despite the simplicity of this concept, the reality of executing three-way matching at scale is far more complicated. Organizations that deal with hundreds or thousands of invoices every month often find themselves buried in paperwork. Finance teams must manually pull up corresponding POs and GRNs, compare line items one by one, and flag any mismatches before approving or rejecting a bill. This process is not only slow but also susceptible to human error, especially when dealing with documents that arrive in a variety of formats such as printed invoices, scanned PDFs, emails, and even handwritten notes [9].

Traditional automation tools have attempted to address this problem through rule-based systems and template matching. These approaches work reasonably well when document formats are consistent and predictable, but they break down when faced with the diversity of real-world financial documents. A slight change in the layout of a supplier's invoice or a different naming convention for line items can cause the entire matching pipeline to fail [1]. As a result, many organizations still rely on significant manual oversight even when automation tools are in place.

The emergence of Large Language Models (LLMs) in recent years has opened up new possibilities for handling unstructured and semi-structured data. LLMs, trained on vast corpora of text, have demonstrated a remarkable ability to understand context, extract relevant information, and even reason over inconsistencies in data [6]. When combined with Optical Character Recognition (OCR) [10], LLMs can process documents in virtually any format and extract the fields necessary for matching, including item names, quantities, unit prices, tax amounts, and totals.

This paper presents a system that applies LLMs to automate the three-way matching process end to end. The system takes in POs, GRNs, and invoices as inputs, extracts structured data using OCR and LLM-based parsing, and then performs matching across the three document types. Discrepancies are flagged with explanations, and bills are classified as either payable or non-payable. The system also generates dashboards and reports that give finance teams clear visibility into the status of their payable pipeline. The primary objective is to reduce the manual workload on AP teams, minimize errors, and make the entire process faster, more transparent, and scalable.

## II. LITERATURE REVIEW

The automation of financial document processing has been a subject of active research for over two decades. Early efforts in this domain focused on using template-based extraction systems, where each document type had a predefined layout and the system would extract data from fixed positions on the page. While these systems were effective in controlled environments, they struggled with the variability of documents encountered in practice. Kaur and Singh [1] provided a comprehensive overview of how AI and OCR technologies were being adopted in accounts payable departments, noting that while OCR had improved significantly, the downstream matching logic remained largely manual or rule-based.

Mishra and Patel [2] examined three-way matching within the context of Enterprise Resource Planning (ERP) systems. Their study highlighted that even with ERP integration, discrepancies between POs, GRNs, and invoices were common and that the resolution of these discrepancies required human judgment. The authors noted that a key limitation of existing ERP-based solutions was their inability to handle unstructured data, such as scanned invoices or handwritten delivery notes.

The application of machine learning to financial document processing gained momentum with the rise of deep learning. Convolutional neural networks and recurrent neural networks were applied to tasks such as document classification, layout analysis, and field extraction [5]. However, these models typically required large amounts of labelled training data and were tailored to specific document types, making them less flexible for generalized AP workflows.

More recently, transformer-based models such as LayoutLM and its successors brought a significant improvement by jointly modelling text, layout, and visual features of documents. These models could understand the spatial relationships between fields on a page, which is particularly useful for extracting data from invoices and receipts where the position of a field matters as much as its text content. Xu et al. [3] demonstrated that LayoutLM achieved state-of-the-art performance on several document understanding benchmarks, including the ICDAR receipt extraction task [7].

The advent of general-purpose LLMs such as GPT-3, GPT-4, and their open-source counterparts added a new dimension to this field. Unlike earlier models that required task-specific fine-tuning, LLMs could be prompted to extract information from documents in a zero-shot or few-shot manner [6]. Palm et al. [4] showed that LLMs could accurately extract structured data from unstructured text when given appropriate instructions. This capability is especially relevant for accounts payable automation, where the variety of document formats makes it impractical to train or fine-tune models for every possible layout. Majumder et al. [8] further demonstrated that representation learning approaches could improve information extraction from form-like documents commonly found in AP workflows.

Despite these advances, the application of LLMs specifically to three-way matching has not been extensively studied. Most existing research focuses on individual subtasks such as invoice data extraction or PO classification, rather than the end-to-end matching pipeline [9]. This paper attempts to bridge that gap by proposing a complete system that uses LLMs for document extraction and matching, while also providing reporting and audit capabilities that are critical for real-world deployment.

## III. PROPOSED SYSTEM

The proposed system is designed as a modular pipeline that takes financial documents as inputs and produces matching results, discrepancy reports, and dashboards as outputs. The system is built around four core modules: document ingestion, data extraction, three-way matching, and reporting. Each module is described in detail below.

### 3.1 Document Ingestion

The ingestion module accepts documents in multiple formats including PDF, scanned images (JPEG, PNG, TIFF), and digital text files. For scanned and image-based documents, the module uses Tesseract OCR [10] to convert images into machine-readable text. For digital PDFs, libraries such as pdfplumber and PyPDF2 are used to extract text directly. The ingestion module normalises the extracted text and passes it to the extraction module.

### 3.2 Data Extraction Using LLMs

The extraction module is the heart of the system. Rather than relying on template-based rules [1], this module sends the extracted text to an LLM along with a structured prompt that instructs the model to identify and return specific fields. For a purchase order, the relevant fields include PO number, vendor name, item descriptions, quantities, unit prices, and the total amount. For a GRN, the fields include GRN number, received quantities, date of receipt, and any notes about damaged or short-shipped goods. For an invoice, the fields include invoice number, billing amounts, tax details, and payment terms.

The prompts are designed using a few-shot approach [6], where a handful of examples are included to guide the LLM on the expected output format. The output is returned as structured JSON, which is then validated and stored in a MySQL database for downstream processing. This approach allows the system to handle a wide variety of document layouts without requiring separate models or templates for each format, a significant advantage over traditional methods [4].

### 3.3 Three-Way Matching Logic

Once the data from all three document types has been extracted and stored, the matching module compares them field by field. The comparison covers several dimensions: item descriptions are compared using string similarity measures and LLM-based semantic matching to account for variations in naming conventions [8]. Quantities are compared numerically, and any difference beyond a configurable tolerance threshold is flagged. Unit prices and total amounts are similarly compared with tolerance for minor rounding differences.

The matching module produces a detailed result for each invoice, indicating whether it is a full match, a partial match, or a mismatch. For partial matches and mismatches, the module provides a breakdown of which fields did not align and the nature of the discrepancy [2]. Based on the overall result and the organization's payment policy, the invoice is classified as payable or non-payable.

### 3.4 Reporting and Dashboards

The final module generates real-time dashboards and audit-ready reports. Dashboards provide a summary view of the payable pipeline, including the number of invoices processed, the match rate, the most common types of discrepancies, and the average processing time. Reports can be generated for individual invoices or in bulk, and include all the details needed for audit and compliance purposes. The dashboards are built using standard visualization libraries and can be integrated with existing business intelligence tools.

## IV. SYSTEM ARCHITECTURE

The system architecture follows a layered design. The presentation layer consists of a web-based dashboard that allows finance users to upload documents, view matching results, and download reports. The application layer contains the core logic for document ingestion, extraction, matching, and reporting. The data layer consists of a MySQL database that stores the extracted data, matching results, and historical records.

At the centre of the application layer is the LLM integration module, which manages communication with the language model API. This module handles prompt construction, response parsing, error handling, and retry logic. It also maintains a cache of recent extractions to avoid redundant API calls for documents that have already been processed.

The OCR module sits alongside the LLM module and is responsible for converting non-digital documents into text. The system uses Tesseract OCR [10] with pre-processing steps such as de-skewing, noise removal, and contrast enhancement to improve recognition accuracy. For documents with complex layouts, such as multi-column invoices or tables, the system uses layout analysis heuristics to segment the page before passing each segment to the OCR engine [3].

The matching engine is implemented as a configurable rule engine that supports both strict and fuzzy matching modes. In strict mode, all fields must match exactly. In fuzzy mode, the engine applies configurable tolerances for numerical fields and uses cosine similarity and LLM-based comparison for text fields [5]. The matching rules can be customized per vendor or per document type, which is useful for organizations that have different tolerance levels for different categories of purchases.

## V. TOOLS AND TECHNOLOGIES

The implementation of the proposed system relies on a combination of open-source tools and widely used Python libraries. The core programming language is Python, chosen for its extensive ecosystem of data processing and machine learning libraries. The key tools and technologies used in the system are described below.

For data manipulation and numerical processing, the system uses pandas and NumPy. These libraries handle the tabular data structures used for storing and comparing extracted fields from POs, GRNs, and invoices. Scikit-learn is used for text similarity calculations and for any classification tasks that arise during the matching process, such as categorizing discrepancy types.

Document parsing is handled by PyPDF2 and pdfplumber for digital PDFs, and by pytesseract (the Python wrapper for Tesseract OCR [10]) for scanned documents and images. These tools convert raw documents into plain text, which is then fed into the LLM-based extraction pipeline.

The LLM component can be configured to use different language models depending on the organization's requirements and budget. The system supports both API-based models (such as those offered by OpenAI [6]) and locally hosted open-source models. Prompts are managed through a templating system that allows easy customization for different document types and fields.

The database layer uses MySQL for storing extracted data, matching results, and audit logs. MySQL was chosen for its reliability, widespread adoption, and ease of integration with Python through libraries such as mysql-connector-python and SQLAlchemy.

The dashboard and reporting module uses standard Python visualization libraries for generating charts and summary statistics. These reports can be exported in PDF or Excel format for offline review.

## VI. IMPLEMENTATION AND RESULTS

### 6.1 Dataset

The system was evaluated using a dataset of purchase orders, goods receipt notes, and invoices. A portion of the data was provided by partnering organizations, while the remainder was simulated to cover a range of edge cases including documents with missing fields, inconsistent formatting, and scanned images with varying quality. The dataset contained a total of 500 document sets, where each set consisted of one PO, one GRN, and one invoice related to the same transaction.

### 6.2 Extraction Accuracy

The LLM-based extraction module was tested on the full dataset. For digitally generated PDFs, the extraction accuracy was above 95% across all key fields including item descriptions, quantities, unit prices, and totals. For scanned documents, the accuracy depended on the quality of the scan; documents with clear print achieved over 90% accuracy, while documents with poor scan quality or handwritten annotations achieved around 80 to 85%. In comparison, a baseline template-based extraction system [1] achieved only 70 to 75% accuracy on the same set of scanned documents, primarily because it could not handle variations in layout.

**Table 1: Extraction Accuracy by Document Type**

| Document Type | LLM-Based Accuracy (%) | Template-Based Accuracy (%) |
|---|---|---|
| Digital PDF (PO) | 96.2 | 92.1 |
| Digital PDF (GRN) | 95.8 | 90.5 |
| Digital PDF (Invoice) | 95.4 | 89.7 |
| Scanned (Clear) | 91.3 | 74.6 |
| Scanned (Poor Quality) | 83.7 | 61.2 |

### 6.3 Matching Results

The three-way matching module was evaluated on the 500 document sets. Out of these, 412 sets (82.4%) were classified as full matches, 53 sets (10.6%) as partial matches, and 35 sets (7.0%) as mismatches. Manual verification of a random sample of 100 results confirmed that the matching accuracy was 94%. The most common types of discrepancies identified were quantity differences (accounting for 38% of all flagged issues), unit price variations (29%), and description mismatches due to different naming conventions used by buyers and suppliers (22%). The remaining 11% involved tax calculation differences and rounding errors.

**Table 2: Three-Way Matching Results Summary**

| Match Category | Count | Percentage (%) |
|---|---|---|
| Full Match | 412 | 82.4 |
| Partial Match | 53 | 10.6 |
| Mismatch | 35 | 7.0 |

### 6.4 Processing Time

One of the significant advantages of the proposed system is the reduction in processing time. On average, the system processed a complete document set (one PO, one GRN, and one invoice) in approximately 12 seconds, including OCR, extraction, matching, and report generation. In contrast, manual processing of the same set by an experienced AP analyst took an average of 8 to 12 minutes. This represents a reduction in processing time of roughly 98%, which translates into substantial time savings when scaled to thousands of invoices per month.

### 6.5 Comparison with Existing Approaches

The proposed system was compared with two baseline approaches: a purely rule-based matching system and a template-based extraction system paired with rule-based matching [1]. The LLM-based system outperformed both baselines across all evaluation metrics. The rule-based system achieved a matching accuracy of 78% but required significant manual configuration for each new vendor or document format. The template-based system achieved 82% accuracy but failed to handle documents that deviated from the predefined templates [2]. The LLM-based system achieved 94% accuracy without requiring any vendor-specific configuration, demonstrating its superiority in handling diverse and unstructured documents [4].

**Table 3: Comparison of Matching Approaches**

| Approach | Accuracy (%) | Vendor Config Required | Handles Unstructured Docs |
|---|---|---|---|
| Rule-Based | 78 | Yes | No |
| Template + Rules | 82 | Yes | Partial |
| LLM-Based (Proposed) | 94 | No | Yes |

## VII. ALIGNMENT WITH SUSTAINABLE DEVELOPMENT GOALS

The proposed system contributes to several United Nations Sustainable Development Goals (SDGs). It aligns with SDG 9 (Industry, Innovation, and Infrastructure) by promoting the adoption of AI-driven automation in financial operations. By making bill processing faster and more reliable, the system supports the digital transformation of enterprise workflows.

The system also aligns with SDG 8 (Decent Work and Economic Growth) by improving the productivity of finance teams. Automating repetitive and error-prone tasks allows employees to focus on higher-value activities such as vendor relationship management, strategic analysis, and process improvement.

Finally, the system contributes to SDG 12 (Responsible Consumption and Production) by minimizing errors and duplicate payments in procurement processes. Accurate three-way matching ensures that organizations only pay for what they actually ordered and received, which reduces financial waste and promotes responsible resource management.

## VIII. CONCLUSION

This paper presented a system for automating the three-way matching process in accounts payable using Large Language Models. The system addresses a long-standing challenge in financial operations [2] by combining OCR-based document processing [10] with LLM-powered data extraction and matching [4] [6]. Unlike traditional template-based or rule-based approaches [1], the proposed system can handle a wide variety of document formats without requiring format-specific configurations, which makes it practical for real-world deployment.

The experimental results showed that the system achieved a matching accuracy of 94% across a dataset of 500 document sets, with an extraction accuracy exceeding 95% for digital documents and above 83% for scanned documents of varying quality. The processing time per document set was reduced from several minutes of manual effort to approximately 12 seconds, representing a significant improvement in efficiency.

Looking ahead, there are several avenues for future work. The extraction accuracy for poor-quality scanned documents could be improved by incorporating more advanced image preprocessing techniques and fine-tuning the OCR engine for specific document types [7]. The matching logic could be enhanced with learning capabilities, where the system adapts its tolerance thresholds over time based on feedback from finance teams. Additionally, integrating the system with ERP platforms such as SAP or Oracle would allow for a more seamless end-to-end workflow from document receipt to payment execution [2].

In summary, the proposed system demonstrates that LLMs offer a practical and scalable approach to automating accounts payable verification, and the results suggest that widespread adoption of such systems could lead to meaningful improvements in the speed, accuracy, and cost-effectiveness of financial operations across industries.

## ACKNOWLEDGMENT

The authors would like to thank the Department of Data Science at Kumaraguru College of Technology, Coimbatore, for providing the resources and guidance necessary for this project. Special thanks to the partnering organizations that provided sample financial documents for evaluation purposes.

**REFERENCES**


[1] Kaur, H., and Singh, J. (2020). Automation in Accounts Payable Using AI and OCR Technology. International Journal of Computer Applications, 175(15), 12-18.

[2] Mishra, S., and Patel, R. (2021). Three-Way Matching in Enterprise Resource Planning Systems: Challenges and Opportunities. Journal of Enterprise Information Management, 34(3), 891-910.

[3] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1192-1200.

[4] Palm, R., Winther, O., and Laws, F. (2022). Information Extraction from Documents Using Large Language Models. Proceedings of the International Conference on Document Analysis and Recognition, 508-522.

[5] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT, 4171-4186.

[6] Brown, T. B., Mann, B., Ryder, N., et al. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

[7] Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., and Jawahar, C. V. (2019). ICDAR 2019 Competition on Scanned Receipt OCR and Information Extraction. Proceedings of the International Conference on Document Analysis and Recognition, 1516-1520.

[8] Majumder, B. P., Potti, N., Tata, S., Wendt, J. B., Zhao, Q., and Najork, M. (2020). Representation Learning for Information Extraction from Form-like Documents. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 6495-6504.

[9] Garnett, N. (2022). Intelligent Document Processing: Transforming Accounts Payable with AI. Journal of Accounting and Finance, 22(4), 45-58.

[10] Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition, 629-633.